withdrawal was made in the Office Action, Applicants assume that such indication of withdrawal is only a typographical error. Correction of this error is respectfully requested.

In the Office Action the Examiner rejected claims 6, 9 and 11 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,023,268 to Britt, Jr. et al. The Examiner rejected, under 35 U.S.C. 103(a), claim 10 as being unpatentable over Britt, Jr. et al., and claims 1-5 and 7-8 as being unpatentable over U.S. Patent No. 6,202,091 to Godse.

The Rejection of Claims 1-5, 7-8 Should Be Withdrawn

The Applicants respectfully request that the Examiner reconsider the rejection with respect to claims 1-5 and 7-8. The Examiner correctly states that Godse does not teach the claimed steps of setting an "archive-commit flag" and a "list-commit flag" on and off, but nonetheless asserted that the status file script described in Godse achieves the same functionality as the claimed flags.

Godse describes a method and apparatus to permit a computer to 'boot up' from its own memory units or from a network when powered up. In case of booting failure, recovery mechanisms allow the computer to revert to an alternative booting scheme. For example, if the local non volatile memory unit has failed or is not available, the computer can receive booting instructions through a network. (Godse, col 5, lines 25-30.) During the booting process a status file is used to maintain status information about the booting process, including which stages have been performed and whether or not they were successful. The file can be examined by the

network manager to determine if all stages of the booting process were executed as specified in a policy file, and to find causes of unexpected failures. (Godse, col. 7, lines 51-59.) The status file simply keeps a log of the booting process, and can be omitted without impairing the booting process. (Godse, col. 7, lines 57-61.)

As described in Godse, the status file is used to record whether a discovery step has been carried out, meaning that the new computer has registered with a system manager through a network. (Godse, col. 7, lines 12-16.) If the discovery is successful, an entry is made in the status file to indicate that discovery has been done, and the computer then awaits file load information through the network. (Godse, col. 10, lines 6-11.) If discovery fails, an entry is made to the status file to indicate that discovery was not done, and the program either aborts or attempts to boot up locally. (Godse, col 10, lines 11-14.) As can be seen from the flow charts in Figs. 5, 8, 10 and 12, the status file is mainly used to keep track of whether discovery was done.

As can be clearly understood from the Godse specification, the status file is used when the computer determines whether to boot up locally or through the network, to find out if the computer successfully registered with the network system manager. Godse does not teach or suggest that the status file be used to indicate whether the storing process for specific data elements received by one computer from a second computer was started, and then to indicate if the storing process was completed. In contrast, claim 1 of the present invention recites "setting an archive-commit flag on" to indicate the beginning of the committing operation, committing the archive to a persistent

storage, and then "setting the archive-commit flag off" to indicate completion of the committing operation. The same process is carried out according to claim 1 with respect to committing to persistent storage a list of required objects, indicating a start of the operation by "setting a list-commit flag on" and indicating completion by "setting the list-commit flag off".

Accordingly, the recording of booting steps in the status file of Godse does not have the same functionality as setting the flags as recited in claim 1. One of ordinary skill in the art would not be motivated by the status file described in Godse to set a flag at the beginning of a persistent storage archive operation, and then to reset that flag at the completion of the operation. Godse's status file does not indicate the start and completion of a storage operation, but instead simply indicates whether a computer should boot from an alternative memory source, such as a network, in case the preferred source, such as a hard disk or non volatile memory, is unavailable.

For at least these reasons, Applicants respectfully submit that claim 1 is not rendered obvious by Godse, and is allowable. Claims 2-5 and 7-8 depend directly or indirectly from claim 1, and at least for that reason are also allowable.

The Rejection of Claims 6, 9, 10 and 11 Should Be Withdrawn

The Applicants respectfully request that the Examiner reconsider the rejection of these claims based on the following. The Examiner asserted that Britt, Jr. et al. teaches the invention

recited in claims 6 and 11, specifically that if the device loses power during downloading of the software, the downloading is later re-established at the point of failure.

Britt, Jr. et al. describes a method to reduce latency while downloading data over a network, that includes an indication of the current block of data being written, so that in case of power loss the downloading can be later resumed from the last block that was written. More specifically, Britt, Jr. et al. describes that the status of a flag indicating loss of power is checked prior to writing any block of downloaded data into flash memory 22b. (Britt, Jr. et al., Col. 12, lines 5-13.) If the flag indicates that power has been interrupted, the writing routine ends. If the flag indicates power is still available, the routine continues writing the next block of data into flash memory. (Britt, Jr. et al., col. 12, lines 13-18.)

A "NUM_BLOCKS" field is provided in flash memory to indicate how many blocks have been written prior to the power being lost. (Britt, Jr. et al., col. 12, lines 18-20.) Once power is restored and a connection to the network is reestablished, the downloading resumes. Since the number of blocks already written is recorded in the flash memory, only the data blocks not already written in memory need to be downloaded, and the download resumes from the last block that was successfully written to flash memory 22b.

The present invention claims a method that is different from what is described in Britt, Jr. et al. In particular, the present invention addresses two aspects of the recovery after loss of power.

The recovery is performed with respect to committing an archive of data to a storage repository, and subsequently with respect to committing a list of objects within that archive to a registry.

As recited in claim 6, a method for initializing a device includes the steps of "determining whether an archive was being committed to persistent storage when said device was powered-off" and, if it is established that the archive was being committed, "instructing said persistent storage to clear the portion of said archive committed to persistent storage." Thus, with respect to the data itself (the archive), claim 6 recites that upon initialization the portion of the archive that was stored before the power failure is cleared.

An example of this process is given in the specification, which states that if the archive storage 40 was not successfully committed to storage, the repository-manager will free any storage blocks being used during step 200 (commit archive to repository), and once device 22 reestablishes communication with server 22, the application begins at step 100 (interrogate device for configuration information) and re-attempts the file transfer from the start. (Specification, P. 8, lines 8-16.)

Accordingly, Applicants respectfully submit that Britt, Jr. et al. does not anticipate the subject matter of claim 6, because Britt, Jr. et al. describes continuing the download of data from where it was interrupted, and does not describe clearing the portion of the archive committed to persistent storage when the device was powered off and restarting the process. Claim 6 is thus allowable. Claims 10 and 11 depend from claim 6, and at least for that reason are also allowable.

Claim 9 addresses committing the list of objects from the archive, rather than the archive itself. Claim 9 recites a method that determines whether the list of objects from the archive was being committed to a registry area of persistent storage when the device was powered off. If the list was being committed, the method of claim 9 continues with "examining said archive to determine a remaining list of objects to be committed to said registry area" and "committing the remainder of said list established from said examining step."

The specification gives an example of this process as well. If device 22 experiences a power failure during step 220 (commit object list to registry), during subsequent initialization gateway G instructs registry 31 to examine the contents of repository 33 to ascertain the list of objects that should be present in registry 31, and use this information to complete step 220. (Specification, p. 8, line 26 to p.9, line 3.)

Therefore, Applicants respectfully submit that Britt, Jr. et al. does not anticipate claim 9, because that reference describes resuming from the point of failure the download of data blocks after a loss of power. Britt, Jr. et al. does not describe determining if the download of a list of objects from an archive was started before the interruption, and completing said list by examining the objects contained in the archive. Accordingly, claim 9 is allowable.

CONCLUSION

In view of the remarks submitted above, the Applicants respectfully submit that the present case is in condition for allowance. All issues raised by the Examiner have been addressed, and a favorable action on the merits is thus earnestly requested.

Respectfully submitted,

Dated: February 8, 2002

By: _____
Patrick J. Fay
Reg. No. 35,508

FAY KAPLUN & MARCIN, LLP
100 Maiden Lane, 17th Floor
New York, NY 10038
(212) 898-8870 (phone)
(212) 208-6819 (facsimile)